

WRDC-TR-90-8007  
Volume VIII  
Part 34

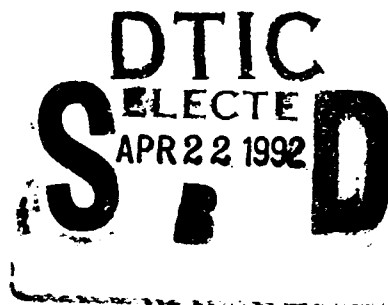
**AD-A248 973**



INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)  
Volume VIII - User Interface Subsystem  
Part 34 - Application Interface Product Specification

S. Barker, F. Glandorf

Control Data Corporation  
Integration Technology Services  
2970 Presidential Drive  
Fairborn, OH 45324-6209



September 1990

Final Report for Period 1 April 1987 - 31 December 1990

Approved for Public Release; Distribution is Unlimited

92-10292

MANUFACTURING TECHNOLOGY DIRECTORATE  
WRIGHT RESEARCH AND DEVELOPMENT CENTER  
AIR FORCE SYSTEMS COMMAND  
WRIGHT-PATTERSON AIR FORCE BASE, OHIO 45433-6533

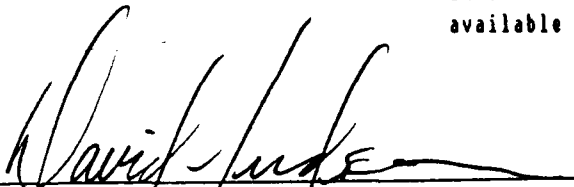
**92 4 21 130**

## NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

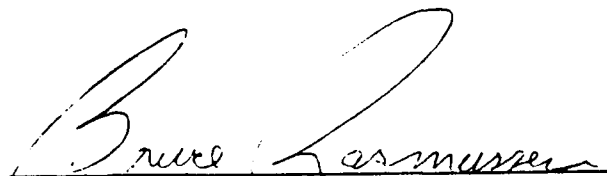
This technical report has been reviewed and is approved for publication.

This report is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations

  
DAVID L. JUDSON, Project Manager  
WRDC/MTI  
Wright-Patterson AFB, OH 45433-6533

25 July 91  
DATE

FOR THE COMMANDER:

  
BRUCE A. RASMUSSEN, Chief  
WRDC/MTI  
Wright-Patterson AFB, OH 45433-6533

25 July 91  
DATE

If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

## REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified			1b. RESTRICTIVE MARKINGS			
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.			
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE						
4. PERFORMING ORGANIZATION REPORT NUMBER(S) PS 620344700			5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR- 90-8007 Vol. VIII, Part 34			
6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services		6b. OFFICE SYMBOL (if applicable) WRDC/MTI	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI			
6c. ADDRESS (City, State, and ZIP Code) 2970 Presidential Drive Fairborn, OH 45324-6209			7b. ADDRESS (City, State, and ZIP Code) WPAFB, OH 45433-6533			
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF		8b. OFFICE SYMBOL (if applicable) WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464			
8c. ADDRESS (City, State, and ZIP Code) Wright-Patterson AFB, Ohio 45433-6533			10. SOURCE OF FUNDING NOS.			
11. TITLE (Include Security Classification) See block 19			PROGRAM ELEMENT NO. 78011F	PROJECT NO. 595600	TASK NO. F95600	WORK UNIT NO. 20950607
12. PERSONAL AUTHOR(S) Structural Dynamics Research Corporation: Barker, S., et al.						
13a. TYPE OF REPORT Final Report		13b. TIME COVERED 4 / 1 / 87 - 12 / 30 / 90	14. DATE OF REPORT (Yr., Mo., Day) 1990 September 30		15. PAGE COUNT 123	
16. SUPPLEMENTARY NOTATION WRDC/MTI Project Priority 6203						
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify block no.)			
FIELD	GROUP	SUB GR.				
1308	0905					
19. ABSTRACT (Continue on reverse if necessary and identify block number)  This specification establishes the detailed design of the Application Interface computer program.  BLOCK 11:  INTEGRATED INFORMATION SUPPORT SYSTEM Vol VIII -User Interface Subsystem  Part 34 - Application Interface Product Specification						
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION Unclassified			
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson			22b. TELEPHONE NO. (Include Area Code) (513) 255-7371		22c. OFFICE SYMBOL WRDC/MTI	

## FOREWORD

This technical report covers work performed under Air Force Contract F33600-87-C-0464, DAPro Project. This contract is sponsored by the Manufacturing Technology Directorate, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Bruce A. Rasmussen, Branch Chief, Integration Technology Division, Manufacturing Technology Directorate, through Mr. David L. Judson, Project Manager. The Prime Contractor was Integration Technology Services, Software Programs Division, of the Control Data Corporation, Dayton, Ohio, under the direction of Mr. W. A. Osborne. The DAPro Project Manager for Control Data Corporation was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test, and demonstration of the Integrated Information Support System (IISS). The IISS technology work comprises enhancements to IISS software and the establishment and operation of IISS test bed hardware and communications for developers and users.

The following list names the Control Data Corporation subcontractors and their contributing activities:

<u>SUBCONTRACTOR</u>	<u>ROLE</u>
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface, and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

	<u>Page</u>
SECTION 1.0 SCOPE .....	1-1
1.1 Identification .....	1-1
1.2 Functional Summary .....	1-1
SECTION 2.0 DOCUMENTS .....	2-1
2.1 Reference Documents .....	2-1
2.2 Terms and Abbreviations .....	2-2
SECTION 3.0 REQUIREMENTS .....	3-1
3.1 Structural Description .....	3-1
3.2 Functional Flow .....	3-2
3.3 Interfaces .....	3-2
3.3.1 Application Programs .....	3-2
3.3.2 Network Transaction Manager .....	3-2
3.3.3 Form Processor .....	3-3
3.4 Program Interrupts .....	3-3
3.5 Timing and Sequencing Description.....	3-3
3.6 Special Control Features .....	3-3
3.7 Storage Allocation .....	3-3
3.8 Object Code Creation .....	3-3
3.9 Adaptation Data .....	3-3
3.10 Detailed Design Description .....	3-3
3.10.1 Main Program List .....	3-3
3.10.2 Module List .....	3-7
3.10.3 External Routines List .....	3-10
3.10.4 Include File List .....	3-12
3.10.5 Where Include File Used List .....	3-14
3.10.6 Where External Routine Used List ....	3-19
3.10.7 Main Program Parts List .....	3-22
3.10.8 Module Documentation .....	3-51
3.10.9 Include File Description .....	3-87
3.10.10 Hierarchy Chart .....	3-96
3.11 Program Listings Comments .....	3-107
SECTION 4.0 QUALITY ASSURANCE PROVISIONS .....	4-1
4.1 Introduction and Definitions .....	4-1
4.2 Computer Programming and Test Evaluation .....	4-1
APPENDIX A FP/AI MESSAGE FORMATS .....	A-1

LIST OF ILLUSTRATIONS

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3-1	AI Interfaces .....	3-1
3-2	AI Data Flow .....	3-2

<b>Accession For</b>	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
<b>By</b>	
Distribution/	
<b>Availability Codes</b>	
<b>Dist</b>	<b>Avail and/or Special</b>
A-1	



## SECTION 1


### SCOPE

#### 1.1 Identification

This specification establishes the detailed design of a computer program identified as the Application Interface, hereinafter referred to as AI. The AI is one configuration item of the Integrated Information Support System (IISS) User Interface (UI).

#### 1.2 Functional Summary

The AI is a collection of procedures that may be linked with an application to enable it to use the Form Processor (FP) and run in the distributed IISS environment. The AI does this by sending/receiving FP requests through the NTM (Network Transaction Manager) to/from the User Interface Monitor (UIM) of the Form Processor.



SECTION 2  
DOCUMENTS

2.1 Reference Documents

- [1] Structural Dynamics Research Corporation, Forms Driven Form Editor Product Specification, PS 620144402 , 1 November 1985.
- [2] Structural Dynamics Research Corporation, Forms Language Compiler Product Specification, PS 620144401 , 1 November 1985.
- [3] Structural Dynamics Research Corporation, Form Processor Product Specification, PS 620144200 , 1 November 1985.
- [4] Structural Dynamics Research Corporation, Rapid Application Generator Product Specification, PS 620144502 , 1 November 1985.
- [5] Structural Dynamics Research Corporation, Report Writer Product Specification, PS 620144501 , 1 November 1985.
- [6] Structural Dynamics Research Corporation, Text Editor Product Specification, PS 620144600 , 1 November 1985.
- [7] Structural Dynamics Research Corporation, User Interface Services Product Specification, PS 620144100 , 1 November 1985.
- [8] Structural Dynamics Research Corporation, Virtual Terminal Product Specification, PS 620144300 , 1 November 1985.
- [9] Structural Dynamics Research Corporation, Application Interface Development Specification, DS 620144700 , 1 November 1985.
- [10] Structural Dynamics Research Corporation, Application Interface Unit Test Plan, UTP620144700 , 1 November 1985.



## 2.2 Terms and Abbreviations

American Standard Code for Information Interchange: (ASCII), the character set defined by ANSI X3.4 and used by most computer vendors.

Application Definition Language: an extension of the Forms Definition Language that includes retrieval of database information and conditional actions. It is used to define interactive application programs.

Application Generator: (AG), subset of the IISS User Interface that consists of software modules that generate IISS application code and associated form definitions based on a language input. The part of the AG that generates report programs is called the Report Writer. The part of the AG that generates interactive applications is called the Rapid Application Generator.

Attribute: field characteristic such as blinking, highlighted, black, etc. and various other combinations. Background attributes are defined for forms or windows only. Foreground attributes are defined for items. Attributes may be permanent, i.e., they remain the same unless changed by the application program, or they may be temporary, i.e., they remain in effect until the window is redisplayed.

Buffer Name: the default file in which the buffer will be saved if no file is given on a save command.

Common Data Model: (CDM), IISS subsystem that describes common data application process formats, form definitions, etc. of the IISS and includes conceptual schema, external schemas, internal schemas, and schema transformation operators.

Conceptual Schema: (CS), the standard definition used for all data in the CDM. It is based on IDEF1 information modelling.

Current Cursor Position: the position of the cursor before an edit command or function is issued in the text editor.

Cursor Position: the position of the cursor after any command is issued.

Cut and Paste Buffer: where deleted lines go and the paste and fill edit commands get their data.

Device Drivers: (DD), software modules written to handle I/O for a specific kind of terminal. The modules map terminal specific commands and data to a neutral format. Device Drivers are part of the UI Virtual Terminal.

Display List: is similar to the open list, except that it contains only those forms that have been added to the screen and are currently displayed on the screen.

Display Start Line: the first line in the buffer to be displayed.

Display Size: the number of lines used in the edit area.

Extended Binary Coded Decimal Interchange Code: (EBCDIC), the character set used by a few computer vendors (notably IBM) instead of ASCII.

External Schema: (ES), an application's view of the CDM's conceptual schema.

Field Pointer: indicates the ITEM which contains the current cursor position.

Forms Driven Form Editor: (FDFE), subset of the FE which consists of a forms driven application used to create Form Definition files interactively.

Form Editor: (FE), subset of the IISS User Interface that is used to create definitions of forms. The FE consists of the Forms Driven Form Editor and the Forms Language Compiler.

Forms Language Compiler: (FLAN), subset of the FE that consists of a batch process that accepts a series of forms definition language statements and produces form definition files as output.

Form Processor Text Editor: (FPTE), subset of the Form Processor that consists of software modules that provide text editing capabilities to all users of applications that use the Form Processor.

Item: non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined areas where user data may be input/output.

Logical Device: a conceptual device which to an application is indistinguishable from a physical device and is then mapped to part or all of a physical device.

Neutral Data Manipulation Language: (NDML), the command language by which the CDM is accessed for the purpose of extracting, deleting, adding, or modifying data.

Open List: a list of all the forms that have been and are currently open for an application process.

Operating System: (OS), software supplied with a computer which allows it to supervise its own operations and manage access to hardware facilities such as memory and peripherals.

Page: instance of forms in windows that are created whenever a form is added to a window.

Paging and Scrolling: a method which allows a form to contain more data than can be displayed with provisions for viewing any portion of the data buffer.

Physical Device: a hardware terminal.

Presentation Schema: (PS), may be equivalent to a form. It is the view presented to the user of the application.

Previous Cursor Position: the position of the cursor when the previous edit command was issued.

Previous Edit Command: the function key pressed before the current one.

Rapid Application Generator: (RAP), part of the Application Generator that generates source code for interactive programs based on a language input.

Report Definition Language: an extension of the Forms Definition Language that includes retrieval and calculation of database information and is used to define reports.

Report Writer: (RW), part of the Application Generator that generates source code for report programs based on a language input.

Select Line: one terminus of the select range.

Select Mode: when on, certain commands will be executed over the lines in the selected range. The commands are <DELETE LINE> and replace.

Subform: a form that is used within another form.

Text Editor: (TE), subset of the IISS User Interface that consists of a file editor that is based on the text editing functions built into the Form Processor.

Top of file: the first line of the buffer.

User Interface Development System: (UIDS), collection of IISS User Interface subsystems that are used by applications programmers as they develop IISS applications. The UIDS includes the Form Editor and the Application Generator.

User Interface Monitor: (UIM), part of the Form Processor that handles messaging between the NTM and the UI. It also provides authorization checks and initiates applications.

User Interface Services: (UIS), subset of the IISS User Interface that consists of a package of routines that aid users in controlling their environment. It includes message management, change password, and application definition services.

Virtual Terminal Interface: (VTI), the callable interface to the VT.

Virtual Terminal Interface Field Map: defines the complete terminal screen by breaking it into pieces of the various forms and items that are displayed. Each area of the terminal screen must be defined as belonging to a particular field in the display list.

Window Manager: a facility which allows the following to be manipulated: size and location of windows, the device on which an application is running, the position of a form within a window. It is part of the Form Processor.

SECTION 3  
REQUIREMENTS

3.1 Structural Description

Applications use the AI just as though they were using the User Interface Form Processor. The Application Interface routines have the same calling sequence as the FP routines, but instead of processing the commands the AI creates messages which are sent to the Form Processor by way of the Network Transaction Manager. The Form Processor then processes the command which is contained in the message. This structure allows the application program to run on a machine other than the host of the User Interface. The detailed structure of the Application Interface is illustrated in section 3.10. Figure 3-1 illustrates the relationship between the AI, NTM, FP and an application.

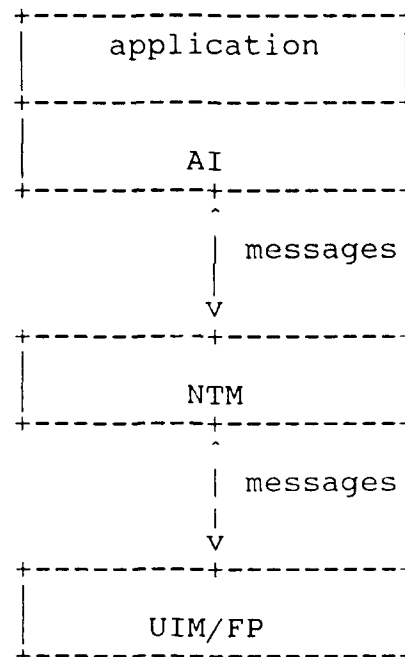


Figure 3-1 AI Interfaces

### 3.2 Functional Flow

Figure 3-2 is a data flow diagram of the Application Interface.

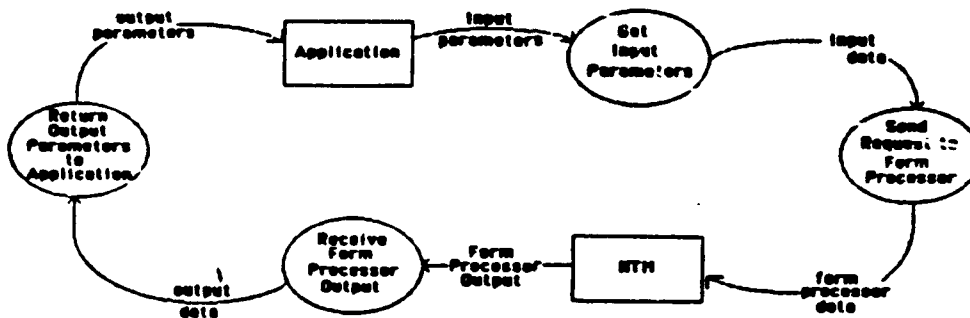


Figure 3-2 Application Interface Data Flow

### 3.3 Interfaces

#### 3.3.1 Application Programs

The interface to the application is identical to that of the Form Processor procedures and is documented in the FP User Manual.

#### 3.3.2 Network Transaction Manager

The AI sends and receives messages to and from the UIM via the NTM. Each FP procedure is identified by a unique number in this message. The remainder of the message consists of FP input parameters for an AI send and FP output parameters for an AI receive. These message formats are documented in Appendix A.

### 3.3.3 Form Processor

The UIM of the Form Processor receives messages from the NTM, calls the appropriate FP procedure and sends the results back to the AI via the NTM.

### 3.4 Program Interrupts

This section does not apply to the detailed design of the Application Interface.

### 3.5 Timing and Sequencing Description

The Application Interface control logic is simple. First, the AI receives input parameters from an application and sends the parameters to the FP via the NTM. Then the AI receives output parameters from the FP via the NTM and returns them to the application. This is illustrated in the flow diagram in section 3.2.

### 3.6 Special Control Features

The detailed design of the Application Interface does not include any special control features as defined in the ICAM Documentation Standards manual.

### 3.7 Storage Allocation

This section does not apply to the AI.

### 3.8 Object Code Creation

The AI routines were compiled with an ANSI COBOL compiler under VAX/VMS. The source is portable to other compilers on machines such as the IBM.

### 3.9 Adaptation Data

The AI source is portable to other ANSI COBOL compilers.

### 3.10 Detailed Design Description

#### 3.10.1 Main Program List

The following is a list of all "Main Programs" which are modules that are not called by any other module being documented here. These modules are either program entry points or, if they are hooked into another set of programs via subroutine calls, they are the points the external programs can call and therefore enter through. To differentiate between the two types of entry points, look at the individual Module Documentation (section 3.10.8) and look at Module Type for each of the Main Program modules listed. Note whether the routine is a Program, Subroutine, or Function. If it is a Program, it

PS 620344700  
30 September 1990

is truly a main program entry point. If not, then it is merely called by other programs not being documented here.



APPLICATION INTERFACE Main Program List

Module Name -----	Purpose -----
ADDELM	ADD ELEMENT
ADDFRM	ADD FORM
CHGLDV	CHANGE LOGICAL DEVICE
CLSFRM	CLOSE FORM
CLSLDV	CLOSE LOGICAL DEVICE
GDATA	GET DATA
GETATT	GET ATTRIBUTES
GETBAK	GET BACKGROUND
GETCUR	GET CURSOR
GPAGE	GET PAGE
GWINDO	GET WINDOW
INITFP	INITialize Form Processor
INITVT	INITIAL VTI
INQLDV	INQUIRE LOGICAL DEVICE
OISCR	OUTPUT / INPUT SCREEN
OPNFRM	OPEN FORM
OPNLDV	OPEN LOGICAL DEVICE
OUTSCR	OUTPUT SCREEN
PARFQN	PARSE FULLY QUALIFIED NAME
PDATA	PUT DATA
PMSGLC	PUT MESSAGE LINE CODE

APPLICATION INTERFACE Main Program List

Module Name -----	Purpose -----
PMSGLS	PUT MESSAGE LINE STRING
PUTATT	PUT ATTRIBUTES
PUTBAK	PUT BACKGROUND
PUTCUR	PUT CURSOR
PUTLOC	PUT CURSOR LOCATION
RMVPAG	REMOVE PAGE
RPLFRM	REPLACE FORM
TERMFP	EXIT FORM PROCESSOR
TERMT	TERMINATE VTI

### 3.10.2 Module List

The following is a list of all the modules being documented here along with their purpose. Each module has a unique name, no matter what language it was written in.

APPLICATION INTERFACE Module List

Module Name -----	Purpose -----
ADDELM	ADD ELEMENT
ADDFRM	ADD FORM
CHGLDV	CHANGE LOGICAL DEVICE
CLSFRM	CLOSE FORM
CLSLDV	CLOSE LOGICAL DEVICE
GDATA	GET DATA
GDATLN	GET DATA LENGTH
GETATT	GET ATTRIBUTES
GETBAK	GET BACKGROUND
GETCUR	GET CURSOR
GETUIM	GET USER INTERFACE MONITOR AP
GPAGE	GET PAGE
GWINDO	GET WINDOW
INITFP	INITialize Form Processor
INITVT	INITIAL VTI
INQLDV	INQUIRE LOGICAL DEVICE
OISCR	OUTPUT / INPUT SCREEN
OPNFRM	OPEN FORM
OPNLDV	OPEN LOGICAL DEVICE
OUTSCR	OUTPUT SCREEN
PARFQN	PARSE FULLY QUALIFIED NAME

APPLICATION INTERFACE Module List

Module Name -----	Purpose -----
PDATA	PUT DATA
PMSGLC	PUT MESSAGE LINE CODE
PMSGLS	PUT MESSAGE LINE STRING
PUTATT	PUT ATTRIBUTES
PUTBAK	PUT BACKGROUND
PUTCUR	PUT CURSOR
PUTLOC	PUT CURSOR LOCATION
RMVPAG	REMOVE PAGE
RPLFRM	REPLACE FORM
TERMFP	EXIT FORM PROCESSOR
TERMTVT	TERMINATE VTI

### 3.10.3 External Routines List

The following is a list of all routines or functions not documented here that are called by modules that are documented here. The first caller, in alphabetical order, is listed as well. See section 3.10.6 for a list of the modules that call each of these external routines.

APPLICATION INTERFACE External Routines List

Module Name -----	First User -----
APACCT	GETUIM
MEMCPY	GETUIM
NSEND	PUTLOC
RCV	TERMVT

#### 3.10.4 Include File List

The following is a list of all include files called in by modules being documented here. Each include file has a unique name regardless of the language being used. The purpose of each include file is listed as well. A more complete description of each include file is given in section 3.10.9. The purpose listed is the one that is in the source code of the include file.

A purpose of "\*\*\*\* PURPOSE NOT FOUND BY STRIPPER \*\*\*\*" indicates that a purpose statement was not written into the include file itself. The most common reason for this is that the include file comes from system libraries that were not developed by the project, such as 'C' libraries that are provided with the 'C' compiler.

See section 3.10.6 for a set of lists which show all the modules which call in each of these include files.



APPLICATION INTERFACE Include File List

File Name -----	Purpose -----
FPCODE	Form Processor return codes
NAPIEVb	CORRECTED C VERSION OF APIEVb.INC
NAPINME	CORRECTED C VERSION OF APINME.INC
NBUFAPi	CORRECTED C VERSION OF BUFAPi.INC
ROUTID	ROUTine ID definitions
SRVRET	AS THE RETURN GIVEN A TABLE-FULL ERROR
STDtYP	STANDARD TYPE DEFINITIONS

#### 3.10.5 Where Include File Used List

The following lists each include file from 3.10.4 and all the modules documented in this specification which include them. The purpose of each module is listed as well.

APPLICATION INTERFACE Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
FPCODE		
	ADDELM	ADD ELEMENT
	ADDFRM	ADD FORM
	CHGLDV	CHANGE LOGICAL DEVICE
	CLSFRM	CLOSE FORM
	CLSLDV	CLOSE LOGICAL DEVICE
	GDATA	GET DATA
	GDATA LN	GET DATA LENGTH
	GETATT	GET ATTRIBUTES
	GETBAK	GET BACKGROUND
	GETCUR	GET CURSOR
	GPAGE	GET PAGE
	GWINDO	GET WINDOW
	INITVT	INITIAL VTI
	INQLDV	INQUIRE LOGICAL DEVICE
	OISCR	OUTPUT / INPUT SCREEN
	OPNFRM	OPEN FORM
	OPNLDV	OPEN LOGICAL DEVICE
	OUTSCR	OUTPUT SCREEN
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDATA	PUT DATA
	PMSGLC	PUT MESSAGE LINE CODE
	PMSGLS	PUT MESSAGE LINE STRING
	PUTATT	PUT ATTRIBUTES
	PUTBAK	PUT BACKGROUND
	PUTCUR	PUT CURSOR
	PUTLOC	PUT CURSOR LOCATION
	RMVPAG	REMOVE PAGE
	RPLFRM	REPLACE FORM
	TERMVT	TERMINATE VTI
NAPIEBV		
	GETUIM	GET USER INTERFACE MONITOR AP

APPLICATION INTERFACE Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
NAPINME	GETUIM	GET USER INTERFACE MONITOR AP
NBUFAPI	GETUIM	GET USER INTERFACE MONITOR AP
ROUTID	ADDELM	ADD ELEMENT
	ADDFRM	ADD FORM
	CHGLDV	CHANGE LOGICAL DEVICE
	CLSFRM	CLOSE FORM
	CLSLDV	CLOSE LOGICAL DEVICE
	GDATA	GET DATA
	GDATA LN	GET DATA LENGTH
	GETATT	GET ATTRIBUTES
	GETBAK	GET BACKGROUND
	GETCUR	GET CURSOR
	GPAGE	GET PAGE
	GWINDO	GET WINDOW
	INITVT	INITIAL VTI
	INQLDV	INQUIRE LOGICAL DEVICE
	OISCR	OUTPUT / INPUT SCREEN
	OPNFRM	OPEN FORM
	OPNLDV	OPEN LOGICAL DEVICE
	OUTSCR	OUTPUT SCREEN
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDATA	PUT DATA
	PMSGLC	PUT MESSAGE LINE CODE
	PMSGLS	PUT MESSAGE LINE STRING
	PUTATT	PUT ATTRIBUTES
	PUTBAK	PUT BACKGROUND
	PUTCUR	PUT CURSOR
	PUTLOC	PUT CURSOR LOCATION
	RMVPAG	REMOVE PAGE

APPLICATION INTERFACE Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
	RPLFRM	REPLACE FORM
	TERMVT	TERMINATE VTI
SRVRET	ADDELM	ADD ELEMENT
	ADDFRM	ADD FORM
	CHGLDV	CHANGE LOGICAL DEVICE
	CLSFRM	CLOSE FORM
	CLSLDV	CLOSE LOGICAL DEVICE
	GDATA	GET DATA
	GDATA LN	GET DATA LENGTH
	GETATT	GET ATTRIBUTES
	GETBAK	GET BACKGROUND
	GETCUR	GET CURSOR
	GPAGE	GET PAGE
	GWINDO	GET WINDOW
	INITVT	INITIAL VTI
	INQLDV	INQUIRE LOGICAL DEVICE
	OISCR	OUTPUT / INPUT SCREEN
	OPNFRM	OPEN FORM
	OPNLDV	OPEN LOGICAL DEVICE
	OUTSCR	OUTPUT SCREEN
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDATA	PUT DATA
	PMSGLC	PUT MESSAGE LINE CODE
	PMSGLS	PUT MESSAGE LINE STRING
	PUTATT	PUT ATTRIBUTES
	PUTBAK	PUT BACKGROUND
	PUTCUR	PUT CURSOR
	PUTLOC	PUT CURSOR LOCATION
	RMVPAG	REMOVE PAGE
	RPLFRM	REPLACE FORM
	TERMVT	TERMINATE VTI

APPLICATION INTERFACE Where-include-file-used List

Include File -----	Module Name -----	Module Purpose -----
STDTP	GETUIM	GET USER INTERFACE MONITOR AP

3.10.6 Where External Routine Used List

The following lists each external function or routine listed in 3.10.3 and all the documented modules which call it. The purpose of each module is listed as well.

APPLICATION INTERFACE Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
APACCT	GETUIM	GET USER INTERFACE MONITOR AP
MEMCPY	GETUIM	GET USER INTERFACE MONITOR AP
NSEND	ADDELM	ADD ELEMENT
	ADDFRM	ADD FORM
	CHGLDV	CHANGE LOGICAL DEVICE
	CLSFRM	CLOSE FORM
	CLSLDV	CLOSE LOGICAL DEVICE
	GDATA	GET DATA
	GDATA LN	GET DATA LENGTH
	GETATT	GET ATTRIBUTES
	GETBAK	GET BACKGROUND
	GETCUR	GET CURSOR
	GPAGE	GET PAGE
	GWINDO	GET WINDOW
	INITVT	INITIAL VTI
	INQLDV	INQUIRE LOGICAL DEVICE
	OISCR	OUTPUT / INPUT SCREEN
	OPNFRM	OPEN FORM
	OPNLDV	OPEN LOGICAL DEVICE
	OUTSCR	OUTPUT SCREEN
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDATA	PUT DATA
	PMSGLC	PUT MESSAGE LINE CODE
	PMSGLS	PUT MESSAGE LINE STRING
	PUTATT	PUT ATTRIBUTES
	PUTBAK	PUT BACKGROUND
	PUTCUR	PUT CURSOR
	PUTLOC	PUT CURSOR LOCATION
	RMVPAG	REMOVE PAGE
	RPLFRM	REPLACE FORM
	TERMVT	TERMINATE VTI



APPLICATION INTERFACE Where-external-routine-used List

System Module -----	Module Name -----	Module Purpose -----
RCV	ADDELM	ADD ELEMENT
	ADDFRM	ADD FORM
	CHGLDV	CHANGE LOGICAL DEVICE
	CLSFRM	CLOSE FORM
	CLSLDV	CLOSE LOGICAL DEVICE
	GDATA	GET DATA
	GDATA LN	GET DATA LENGTH
	GETATT	GET ATTRIBUTES
	GETBAK	GET BACKGROUND
	GETCUR	GET CURSOR
	GPAGE	GET PAGE
	GWINDO	GET WINDOW
	INITVT	INITIAL VTI
	INQLDV	INQUIRE LOGICAL DEVICE
	OISCR	OUTPUT / INPUT SCREEN
	OPNFRM	OPEN FORM
	OPNLDV	OPEN LOGICAL DEVICE
	OUTSCR	OUTPUT SCREEN
	PARFQN	PARSE FULLY QUALIFIED NAME
	PDATA	PUT DATA
	PUTATT	PUT ATTRIBUTES
	PUTBAK	PUT BACKGROUND
	PUTCUR	PUT CURSOR
	PUTLOC	PUT CURSOR LOCATION
	RMVPAG	REMOVE PAGE
	RPLFRM	REPLACE FORM
	TERMVT	TERMINATE VTI

### 3.10.7 Main Program Parts List

The following lists each Main Program listed in 3.10.1 and all the modules which are called either by that module itself or by any of the documented modules which it calls. It is possible for a non-main module to be listed more than once if it is called by multiple modules. The called modules, in this case known as program parts, are marked as to whether they are documented here. If so, the phrase "well-defined module" appears by the module name, if not it is an "external routine". The Purpose of the Main Program module is listed as well.

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
ADDELM		Purpose-->ADD ELEMENT
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
ADDFRM		Purpose-->ADD FORM
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
CHGLDV		Purpose-->CHANGE LOGICAL DEVICE
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
CLSFRM		Purpose-->CLOSE FORM
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
CLSLDV		Purpose-->CLOSE LOGICAL DEVICE
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
GDATA		Purpose-->GET DATA
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine



APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
GETATT		Purpose-->GET ATTRIBUTES
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
GETBAK		Purpose-->GET BACKGROUND
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
GETCUR		Purpose-->GET CURSOR
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
GPAGE		Purpose-->GET PAGE
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
GWINDO		Purpose-->GET WINDOW
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
INITVT		Purpose-->INITIAL VTI
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
INQLDV		Purpose-->INQUIRE LOGICAL DEVICE
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
OISCR		Purpose-->OUTPUT / INPUT SCREEN
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine



APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
OPNFRM		Purpose-->OPEN FORM
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
OPNLDV		Purpose-->OPEN LOGICAL DEVICE
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
OUTSCR		Purpose-->OUTPUT SCREEN
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PARFQN		Purpose-->PARSE FULLY QUALIFIED NAME
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PDATA		Purpose-->PUT DATA
	APACCT	External routine
	GDATLN	Well-defined module
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PMSGLC		Purpose-->PUT MESSAGE LINE CDF
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PMSGLS		Purpose-->PUT MESSAGE LINE STRING
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PUTATT		Purpose-->PUT ATTRIBUTES
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine



APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PUTBAK		Purpose-->PUT BACKGROUND
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PUTCUR		Purpose-->PUT CURSOR
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
PUTLOC		Purpose-->PUT CURSOR LOCATION
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
RMVPAG		Purpose-->REMOVE PAGE
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
RPLFRM		Purpose-->REPLACE FORM
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

APPLICATION INTERFACE Main Program Parts List

Main Pgm Name -----	Module Name -----	Module Type -----
TERMVT		Purpose-->TERMINATE VTI
	APACCT	External routine
	GETUIM	Well-defined module
	MEMCPY	External routine
	NSEND	External routine
	RCV	External routine

### 3.10.8 Module Documentation

The following documentation describes information which is specific to each individual module being documented in this specification as listed in section 3.10.2. It provides a compact way of getting information that would be otherwise buried within each module's source code.

The specific items in this module documentation have the following meanings:

NAME:	Name of program Module.
PURPOSE:	Purpose of Module as detailed in the source code.
LANGUAGE:	Programming language source code is written in. The choices are: C VAX-11 FORTRAN (I/S-1 Workbench 'C') VAX-11 COBOL
MODULE TYPE:	Whether a Program, Subroutine, or Function.
SOURCE FILE:	Name of Source File from file specification.
SOURCE FILE TYPE:	Source File Extension from file specification.
HOST:	Whether this is a host-dependent routine (VAX or IBM) or blank if host-independent.
SUBSYSTEM:	IISS sub-system this file resides in.
SUBDIRECTORY:	Sub-directory of that subsystem in which this file resides.
DOCUMENTATION GROUP:	Name of documentation group of which this source file is a member.
DESCRIPTION:	A description of the module as obtained from the source code.
ARGUMENTS:	The arguments with which this routine is called if it is a Subroutine or a Function.
INCLUDE FILES:	A list of all the files that are included into this module as well as their purposes.

ROUTINES CALLED: Subroutines or Functions, either documented or external, called by this module, if any.

CALLED DIRECTLY BY: The documented routines which call this module, if any.

USED IN MAIN PROGRAM(S): The documented Main Programs which contain this module in their parts list according to the list in section 3.10.7.

The Module Documentation is arranged alphabetically according to Module Name.



APPLICATION INTERFACE Module Documentation

NAME: ADDELM  
PURPOSE: ADD ELEMENT  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: ADDELM

DESCRIPTION:

-----

INCLUDE FILES:

-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: ADDFRM  
PURPOSE: ADD FORM  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: ADDFRM

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: CHGLDV  
PURPOSE: CHANGE LOGICAL DEVICE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: CHGLDV

DESCRIPTION:

-----

INCLUDE FILES:

-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: CLSFRM  
PURPOSE: CLOSE FORM  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: CLSFRM

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: CLSLDV  
PURPOSE: CLOSE LOGICAL DEVICE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: CLSLDV

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

# APPLICATION INTERFACE Module Documentation

NAME: GDATA  
PURPOSE: GET DATA  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: GDATA

DESCRIPTION:  
-----

## INCLUDE FILES:

-----  
FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

## ROUTINES CALLED:

-----  
GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: GDATLN  
PURPOSE: GET DATA LENGTH  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: GDATLN

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

CALLED DIRECTLY BY:  
-----

PDATA - PUT DATA

USED IN MAIN PROGRAM(S):  
-----

PDATA - PUT DATA

# APPLICATION INTERFACE Module Documentation

NAME: GETATT  
PURPOSE: GET ATTRIBUTES  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: GETATT

DESCRIPTION:  
-----

## INCLUDE FILES:

-----  
FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

## ROUTINES CALLED:

-----  
GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV



APPLICATION INTERFACE Module Documentation

NAME: GETBAK  
PURPOSE: GET BACKGROUND  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: GETBAK

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: GETCUR  
PURPOSE: GET CURSOR  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: GETCUR

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

# APPLICATION INTERFACE Module Documentation

NAME: GETUIM  
PURPOSE: GET USER INTERFACE MONITOR AP  
LANGUAGE: C  
MODULE TYPE: FUNCTION  
FUNCTION TYPE: FORTRAN VOID ()  
SOURCE FILE: GETUIM  
SOURCE FILE TYPE: .C  
HOST:  
SUBSYSTEM: UI  
SUBDIRECTORY: FPAI  
DOCUMENTATION GROUP: FPAI

## DESCRIPTION:

-----

### SYNOPSIS

```
FORTRAN VOID GETUIM(UIAPNM, UICHAN, RCODE)
CHAR UIAPNM[APNAM_LEN], UICHAN[LCHAN_LEN],
RCODE[RCODE_LEN];
```

### OUTPUTS:

UIAPNM - AP NAME OF UIM  
UICHAN - LOGICAL CHANNEL FOR UIM  
RCODE - RETURN CODE

### DESCRIPTION

RETURNS THE AP NAME AND LOGICAL CHANNEL OF THE USER  
INTERFACE MONITOR.

### ARGUMENTS:

-----

UIAPNM = CHAR [APNAM\_LEN ]  
UICHAN = CHAR [LCHAN\_LEN ]  
RCODE = CHAR [RCODE\_LEN ]

### INCLUDE FILES:

-----

!TDTYP - STANDARD TYPE DEFINITIONS  
!APINME - CORRECTED C VERSION OF APINME.INC  
!BUFAPI - CORRECTED C VERSION OF BUFAPI.INC  
!APIEBV - CORRECTED C VERSION OF APIEBV.INC

### ROUTINES CALLED:

-----

APACCT  
MEMCPY

### CALLED DIRECTLY BY:

-----

ADDELM - ADD ELEMENT

ADDFRM	- ADD FORM
CHGLDV	- CHANGE LOGICAL DEVICE
CLSFRM	- CLOSE FORM
CLSLDV	- CLOSE LOGICAL DEVICE
GDATA	- GET DATA
GDATA LN	- GET DATA LENGTH
GETATT	- GET ATTRIBUTES
GETBAK	- GET BACKGROUND
GETCUR	- GET CURSOR
GPAGE	- GET PAGE
GWINDO	- GET WINDOW
INITVT	- INITIAL VTI
INQLDV	- INQUIRE LOGICAL DEVICE
OISCR	- OUTPUT / INPUT SCREEN
OPNFRM	- OPEN FORM
OPNLDV	- OPEN LOGICAL DEVICE
OUTSCR	- OUTPUT SCREEN
PARFQN	- PARSE FULLY QUALIFIED NAME
PDATA	- PUT DATA
PMSGLC	- PUT MESSAGE LINE CODE
PMSGLS	- PUT MESSAGE LINE STRING
PUTATT	- PUT ATTRIBUTES
PUTBAK	- PUT BACKGROUND
PUTCUR	- PUT CURSOR
PUTLOC	- PUT CURSOR LOCATION
RMVPAG	- REMOVE PAGE
RPLFRM	- REPLACE FORM
TERMT	- TERMINATE VTI

USED IN MAIN PROGRAM(S):

ADDELM	- ADD ELEMENT
ADDFRM	- ADD FORM
CHGLDV	- CHANGE LOGICAL DEVICE
CLSFRM	- CLOSE FORM
CLSLDV	- CLOSE LOGICAL DEVICE
GDATA	- GET DATA
GETATT	- GET ATTRIBUTES
GETBAK	- GET BACKGROUND
GETCUR	- GET CURSOR
GPAGE	- GET PAGE
GWINDO	- GET WINDOW
INITVT	- INITIAL VTI
INQLDV	- INQUIRE LOGICAL DEVICE
OISCR	- OUTPUT / INPUT SCREEN
OPNFRM	- OPEN FORM
OPNLDV	- OPEN LOGICAL DEVICE
OUTSCR	- OUTPUT SCREEN
PARFQN	- PARSE FULLY QUALIFIED NAME
PDATA	- PUT DATA
PMSGLC	- PUT MESSAGE LINE CODE
PMSGLS	- PUT MESSAGE LINE STRING
PUTATT	- PUT ATTRIBUTES
PUTBAK	- PUT BACKGROUND
PUTCUR	- PUT CURSOR
PUTLOC	- PUT CURSOR LOCATION

PS 620344700  
30 September 1990

RMVPAG	- REMOVE PAGE
RPLFRM	- REPLACE FORM
TERMVT	- TERMINATE VTI

APPLICATION INTERFACE Module Documentation

NAME: GPAGE  
PURPOSE: GET PAGE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: GPAGE

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: GWINDO  
PURPOSE: GET WINDOW  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: GWINDO

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME:	INITFP
PURPOSE:	INITialize Form Processor
LANGUAGE:	VAX-11 COBOL
MODULE TYPE:	PROGRAM
SOURCE FILE:	INITFP

DESCRIPTION:

-----



APPLICATION INTERFACE Module Documentation

NAME: INITVT  
PURPOSE: INITIAL VTI  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: INITVT

DESCRIPTION:

-----

INCLUDE FILES:

-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: INQLDV  
PURPOSE: INQUIRE LOGICAL DEVICE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: INQLDV

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: OISCR  
PURPOSE: OUTPUT / INPUT SCREEN  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: OISCR

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: OPNFRM  
PURPOSE: OPEN FORM  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: OPNFRM

DESCRIPTION:

-----

INCLUDE FILES:

-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: OPNLDV  
PURPOSE: OPEN LOGICAL DEVICE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: OPNLDV

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: OUTSCR  
PURPOSE: OUTPUT SCREEN  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: OUTSCR

DESCRIPTION:

-----

INCLUDE FILES:

-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: PARFQN  
PURPOSE: PARSE FULLY QUALIFIED NAME  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: PARFQN

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: PDATA  
PURPOSE: PUT DATA  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: PDATA

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GDATA LN - GET DATA LENGTH  
GETUIM - GET USER INTERFACE MCNITOR AP  
NSEND  
RCV



APPLICATION INTERFACE Module Documentation

NAME: PMSGLC  
PURPOSE: PUT MESSAGE LINE CODE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: PMSGLC

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND

APPLICATION INTERFACE Module Documentation

NAME: PMSGLS  
PURPOSE: PUT MESSAGE LINE STRING  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: PMSGLS

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND

# APPLICATION INTERFACE Module Documentation

NAME: PUTATT  
PURPOSE: PUT ATTRIBUTES  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: PUTATT

## DESCRIPTION:

## INCLUDE FILES:

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

## ROUTINES CALLED:

GETUIM - GET USER INTERFACE MONITOR AP  
HSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: PUTBAK  
PURPOSE: PUT BACKGROUND  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: PUTBAK

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: PUTCUR  
PURPOSE: PUT CURSOR  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: PUTCUR

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

# APPLICATION INTERFACE Module Documentation

NAME: PUTLOC  
PURPOSE: PUT CURSOR LOCATION  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: PUTLOC

DESCRIPTION:  
-----

## INCLUDE FILES:

-----  
FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

## ROUTINES CALLED:

-----  
GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: RMVPAG  
PURPOSE: REMOVE PAGE  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: RMVPAG

DESCRIPTION:  
-----

INCLUDE FILES:  
-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:  
-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

APPLICATION INTERFACE Module Documentation

NAME: RPLFRM  
PURPOSE: REPLACE FORM  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: RPLFRM

DESCRIPTION:

-----

INCLUDE FILES:

-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV



APPLICATION INTERFACE Module Documentation

NAME:	TERMFP
PURPOSE:	EXIT FORM PROCESSOR
LANGUAGE:	VAX-11 COBOL
MODULE TYPE:	PROGRAM
SOURCE FILE:	TERMFP

DESCRIPTION:

-----

APPLICATION INTERFACE Module Documentation

NAME: TERMVT  
PURPOSE: TERMINATE VTI  
LANGUAGE: VAX-11 COBOL  
MODULE TYPE: PROGRAM  
SOURCE FILE: TERMVT

DESCRIPTION:

-----

INCLUDE FILES:

-----

FPCODE - Form Processor return codes  
ROUTID - ROUTine ID definitions  
SRVRET - AS THE RETURN GIVEN A TABLE-FULL ERROR

ROUTINES CALLED:

-----

GETUIM - GET USER INTERFACE MONITOR AP  
NSEND  
RCV

#### 3.10.9 Include File Descriptions

The following list contains a purpose and description of each include file listed in 3.10.4 as specified in the source code. The language it is written in is also given.

APPLICATION INTERFACE Include File Description

FILE NAME: FPCODE  
PURPOSE: Form Processor return codes  
LANGUAGE: VAX-11 COBOL

DESCRIPTION:  
-----

APPLICATION INTERFACE Include File Description

FILE NAME: NAPIEBV  
PURPOSE: CORRECTED C VERSION OF APIEBV.INC  
LANGUAGE: C

DESCRIPTION:  
-----

PS 620344700  
30 September 1990

APPLICATION INTERFACE Include File Description

FILE NAME: NAPINME  
PURPOSE: CORRECTED C VERSION OF APINME.INC  
LANGUAGE: C

DESCRIPTION:  
-----

DESCRIPTION  
APIS GLOBAL NAME DATA.

APPLICATION INTERFACE Include File Description

FILE NAME: NBUFAPI  
PURPOSE: CORRECTED C VERSION OF BUFAPI.INC  
LANGUAGE: C

DESCRIPTION:  
-----

APPLICATION INTERFACE Include File Description

FILE NAME: ROUTID  
PURPOSE: ROUTine ID definitions  
LANGUAGE: VAX-11 COBOL

DESCRIPTION:  
-----

DESCRIPTION: THIS INCLUDE MEMBER CONTAINS THE ROUTINE ID  
DEFINITIONS FOR MESSAGING BETWEEN THE FPAI  
AND THE FP.

INFORMATION:

TYPE: (C-COBOL, IC-COBOL COPY) IC  
SUBSYSTEM: UI  
CONFIGURATION ITEM ID:

DESIGNED BY: A. J. WEHRMAN  
START DATE: 8/8/85  
FINISH DATE: 8/8/85

PROGRAMMED BY: A. J. WEHRMAN  
START DATE: 8/8/85  
FINISH DATE: 8/8/85



APPLICATION INTERFACE Include File Description

FILE NAME: SRVRET  
PURPOSE: AS THE RETURN GIVEN A TABLE-FULL ERROR  
LANGUAGE: VAX-11 COBOL

DESCRIPTION:  
-----

MODIFIED 11/2/83 TO INCLUDE RET-CODE-5	*
MODIFIED 1/9/84 TO INCREASE ALL ERROR CODES TO PIC X(5)	*
AND TO ELIMINATE ALPHA'S	*
MODIFIED 1/26/84 TO ADD RET-CODE FOR GETUSR-NOT-SUCC	*
SRV-SUCCESSFUL ADDED FOR GENERIC RETURN	*
MODIFIED 2/7/84 TO ADD ERROR CODES FOR ENTRY-NOT-FOUND	*
MODIFIED 2/8/84 TO ADD WTHST-NOT-SUCCESSFUL	*
MODIFIED 2/20/84 TO ADD TSTMOD NEW CODES.	*
MODIFIED 20 AUG 84 INITIALIZE ALL LOCAL VARIABLES TO SPACES OR 0.	
MODIFIED 5/21/85 TO ADD RCL AND FILGEN RETURN CODES	

APPLICATION INTERFACE Include File Description

FILE NAME: STDTyp  
PURPOSE: STANDARD TYPE DEFINITIONS  
LANGUAGE: C

DESCRIPTION:  
-----

DESCRIPTION

THIS FILE ENSURES THAT THE FOLLOWING STANDARD TYPES ARE  
AVAILABLE:

Float	- SINGLE PRECISION Float
DOUBLE	- DOUBLE PRECISION Float
LONG	- 32 BIT (OR LARGER) SIGNED INTEGER
LBITS	- 32 BITS (OR MORE) FOR BIT MANIPULATION
INT	- NATURAL SIZE SIGNED INTEGER
UNSIGNED	- NATURAL SIZE UNSIGNED INTEGER
BOOL	- NATURAL SIZE LOGICAL (ZERO / NON-ZERO ONLY)
SHORT	- 16 BIT (OR LARGER) SIGNED INTEGER
USHORT	- 16 BIT (OR LARGER) UNSIGNED INTEGER
BITS	- 16 BITS (OR MORE) FOR BIT MANIPULATION
CHAR	- SINGLE MACHINE CHARACTER (REAL CHARACTERS ALWAYS POSITIVE)
TINY	- 8 BIT (OR LARGER) SIGNED INTEGER
UTINY	- 8 BIT (OR LARGER) UNSIGNED INTEGER
TBITS	- 8 BITS (OR MORE) FOR BIT MANIPULATION
TBOOL	- 8 BIT (OR LARGER) LOGICAL (ZERO / NON-ZERO ONLY)
METACHAR	- 16 BIT (OR LARGER) AUGMENTED CHARACTER (SIGNED)
VOID	- FUNCTION THAT RETURNS NO VALUE
FORTTRAN	- STORAGE CLASS FOR FOREIGN (NON-C) ROUTINES OR C ROUTINES WHICH ARE CALLABLE FROM FOREIGN ROUTINES

SINCE NOT ALL COMPILERS SUPPORT USHORT, TINY, AND UTINY,  
THE FUNCTIONS  
USHORT(), TINY(), AND UTINY() SHOULD BE USED WHENEVER  
REFERENCING THEM.

IN ADDITION, THE FOLLOWING UTILITY MACROS ARE DEFINED:  
LURSHIFT(N, B) - UNSIGNED LONG RIGHT SHIFT  
MAX(A, B) - MAXIMUM OF A AND B  
MIN(A, B) - MINIMUM OF A AND B

APPLICATION INTERFACE Include File Description

ABS(A) - ABSOLUTE VALUE OF A  
STRASN(A, B) - TRANSPORTABLE A = B FOR STRUCTURES  
NULL - NULL POINTER VALUE (0)  
TRUE - 1  
FALSE - 0  
SUCCESS - EXIT(SUCCESS) INDICATES SUCCESSFUL  
COMPLETION  
FAILURE - EXIT(FAILURE) INDICATES ERRORS

THE FOLLOWING SYMBOLS SHOULD BE DEFINED BASED ON THE  
COMPILER BEING USED:

USHORT - COMPILER SUPPORTS UNSIGNED SHORT  
TINY - COMPILER TREATS CHAR AS SIGNED  
UTINY - CHAR IS SIGNED AND COMPILER SUPPORTS  
UNSIGNED CHAR  
VOID - COMPILER SUPPORTS VOID  
FORTRAN - COMPILER SUPPORTS FORTRAN  
STRASN - DEFINE APPROPRIATE MACRO  
SUCCESS - DEFINE APPROPRIATE VALUE IF NOT 0  
FAILURE - DEFINE APPROPRIATE VALUE IF NOT 1

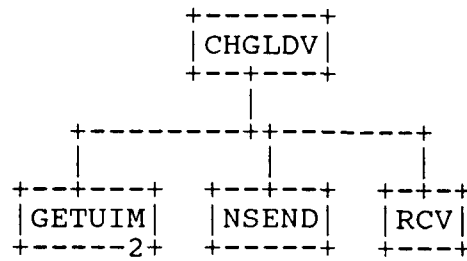
### 3.10.10 Hierarchy Chart

The following hierarchy charts show the relationships between all of the modules mentioned in the above documentation. A module may call a subroutine several times within its code, but the call will only be shown once as a single relationship on this hierarchy chart. All modules shown at the top of the first page are considered Main Programs as described in section 3.10.1 above.

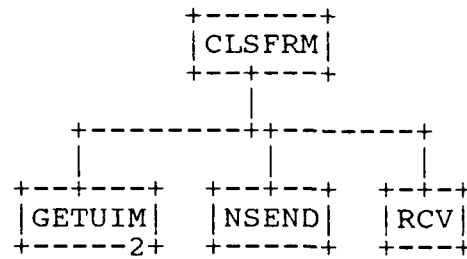
There is an internal paging scheme as marked by the numbers in the upper right corner of each page. An index after the last page of the chart shows where a routine and its calls are first defined. If a routine has no page reference, it either makes no calls or is an external routine. A continuation box on the end of a tree limb shows where that the tree continues on the page numbered mentioned. A number in a box with a routine name points to the page where the routine is further defined within the hierarchy tree. If there is no number in a box, the routine either makes no calls or is an external routine.



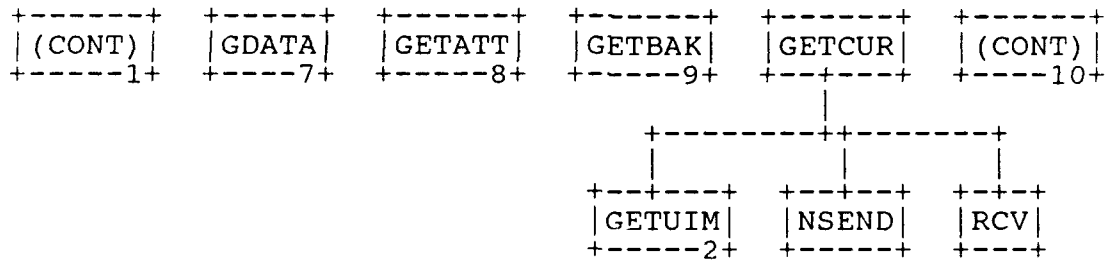
4



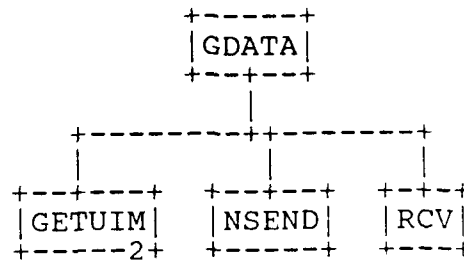
5



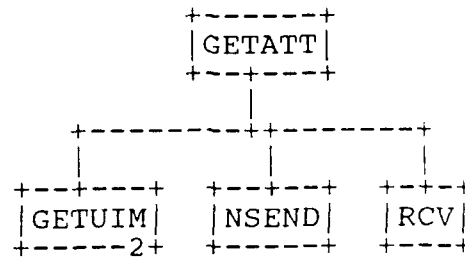
6



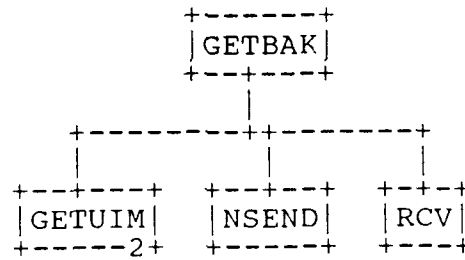
7

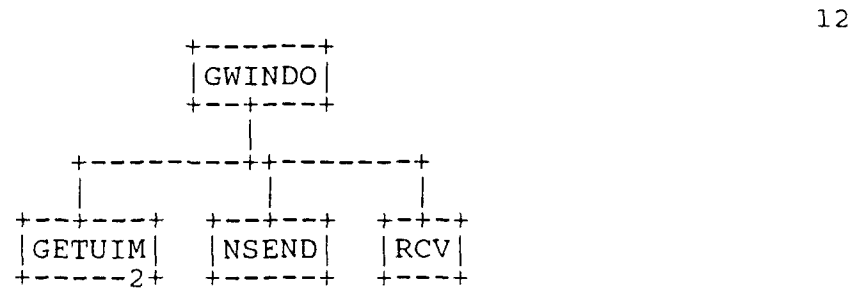
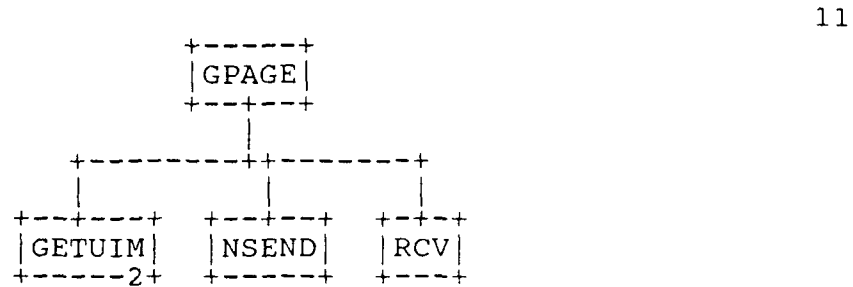
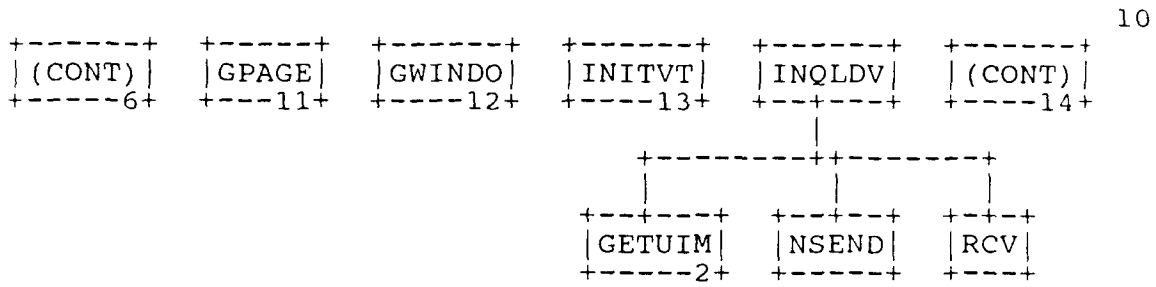


8



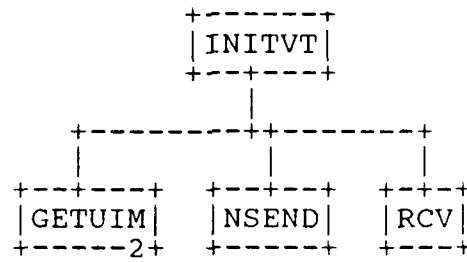
9



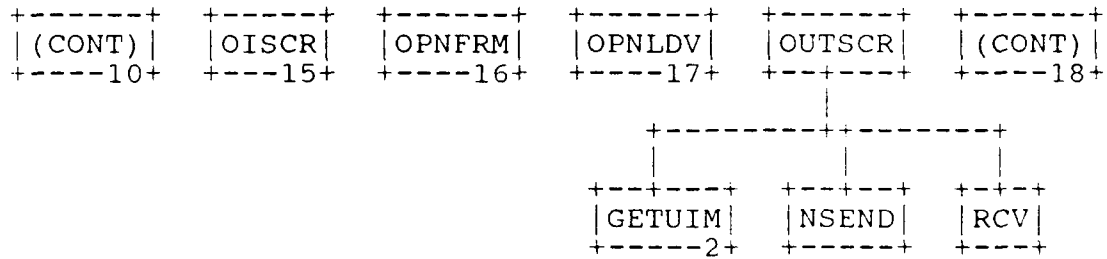




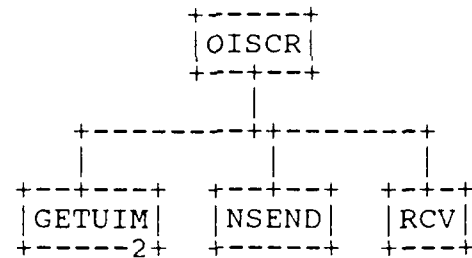
13



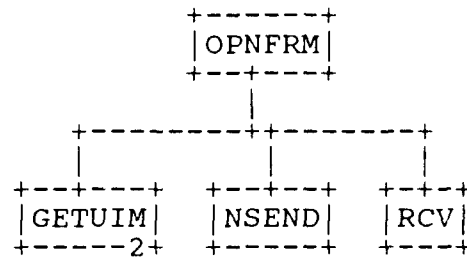
14



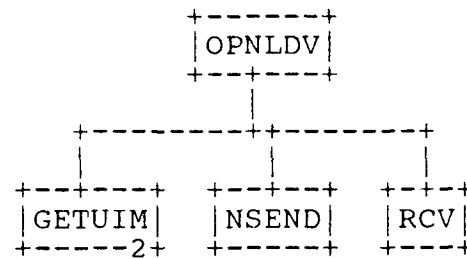
15



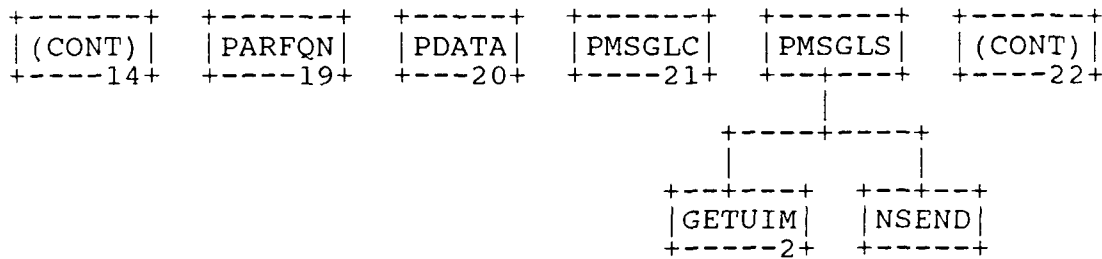
16



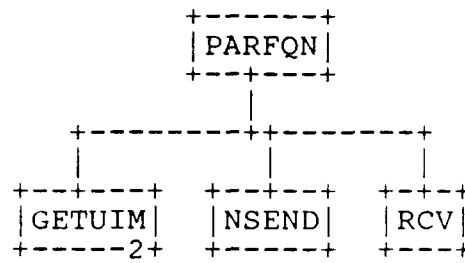
17



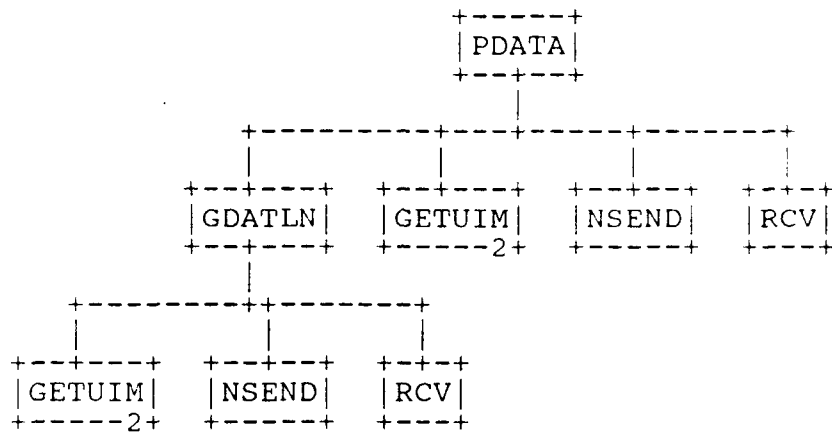
18



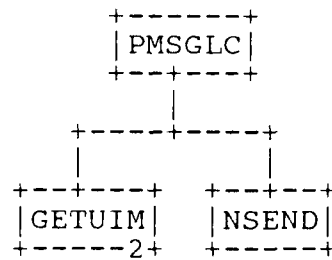
19



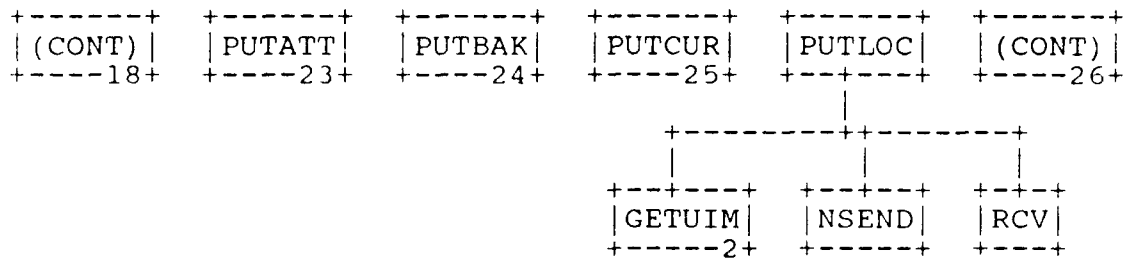
20



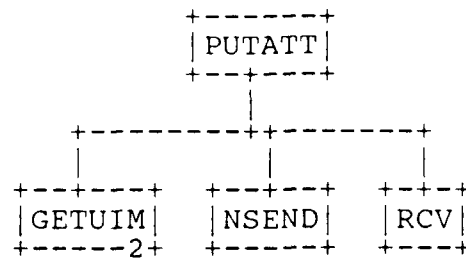
21



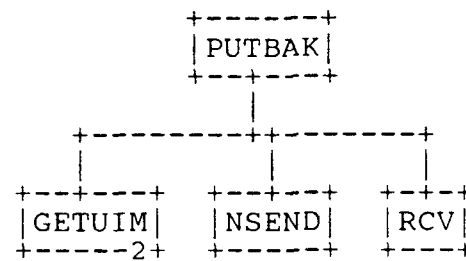
22



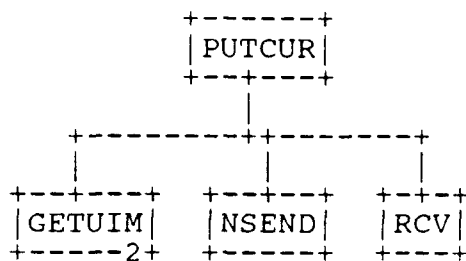
23



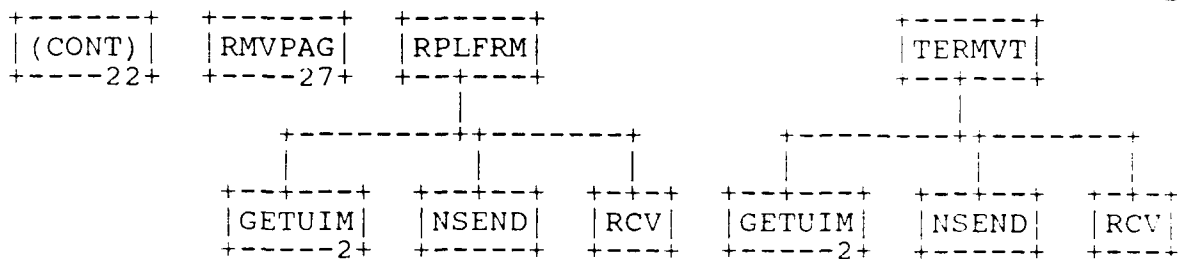
24



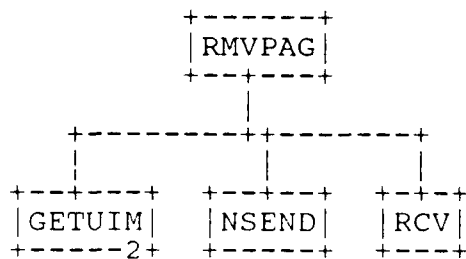
25



26



27



ADDELM....2  
ADDFRM....3  
APACCT  
CHGLDV....4  
CLSFRM....5  
CLSLDV....1  
GDATA.....7  
GDATLN...20  
GETATT....8  
GETBAK....9  
GETCUR....6  
GETUIM....2  
GPAGE....11  
GWINDO...12  
INITVT...13  
INQLDV...10  
MEMCPY  
NSEND  
OISCR....15  
OPNFRM...16  
OPNLDV...17  
OUTSCR...14  
PARFQN...19  
PDATA....20  
PMSGLC...21  
PMSGLS...18  
PUTATT...23  
PUTBAK...24  
PUTCUR...25  
PUTLOC...22  
RCV  
RMVPAG...27  
RPLFRM...26  
TERMVT...26

### 3.11 Program Listings Comments

This information is contained in the Module Descriptions in section 3.10.

## SECTION 4

### QUALITY ASSURANCE PROVISIONS

#### 4.1 Introduction and Definitions

"Testing" is a systematic process that may be preplanned and explicitly stated. Test techniques and procedures may be defined in advance, and a sequence of test steps may be specified. "Debugging" is the process of isolation and correction of the cause of an error.

"Antibugging" is defined as the philosophy of writing programs in such a way as to make bugs less likely to occur and when they do occur, to make them more noticeable to the programmer and the user. In other words, as much error checking as is practical and possible in each routine should be performed.

#### 4.2 Computer Programming Test and Evaluation

The quality assurance provisions for test consists of the normal testing techniques that are accomplished during the construction process. They consist of design and code walk-throughs, unit testing, and integration testing. These tests are performed by the design team. Structured design, design walk-through and the incorporation of "antibugging" facilitate this testing by exposing and addressing problem areas before they become coded "bugs."



APPENDIX A

FP/AI MESSAGE FORMATS

The following is a COBOL declaration of the message formats used between the AI and the FP.

FP to AI message formats, FP output parameters.

NOTE: All 01 line records correspond to the FP routines being called. For example, ADDELM-FP-RECORD is used in the routine ADDELM.

```
01  ADDELM-FP-RECORD.
    05  ADDELM-ELEMENT-NUMBER    PIC 9(4).
    05  ADDELM-RCODE             PIC X(5).
01  ADDFRM-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05  ADDFRM-PAGE-NUMBER       PIC 9(4).
    05  ADDFRM-RCODE             PIC X(5).
01  CHGLDV-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05  CHGLDV-RCODE             PIC X(5).
01  CLSFRM-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05  CLSFRM-RCODE             PIC X(5).
01  CLSLDV-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05  CLSLDV-RCODE             PIC X(5).
01  GDATA-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05  GDATA-BUFFER-LENGTH      PIC 9(4).
    05  GDATA-RCODE              PIC X(5).
    05  GDATA-BUFFER             PIC X(4096).
01  GDATLN-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05  GDATLN-BUFFER-LENGTH     PIC 9(4).
    05  GDATLN-RCODE             PIC X(5).
01  GETATT-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05  GETATT-ATTRIBUTE         PIC X(10).
    05  GETATT-RCODE             PIC X(5).
01  GETBAK-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05  GETBAK-ATTRIBUTE         PIC X(10).
    05  GETBAK-RCODE             PIC X(5).
01  GETCUR-FP-RECORD REDEFINES ADDELM-FP-RECORD.
    05  GETCUR-FIELD-NAME        PIC X(120).
    05  GETCUR-FIELD-TYPE        PIC X.
    05  GETCUR-ROW               PIC 9(4).
    05  GETCUR-COL               PIC 9(4).
    05  GETCUR-RCODE             PIC X(5).
```

```
01 GPAGE-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 GPAGE-FORM-NAME PIC X(10).
05 GPAGE-RCODE PIC X(5).
01 GWINDO-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 GWINDO-TOTAL-PAGES PIC 9(4).
05 GWINDO-RCODE PIC X(5).
01 INQLDV-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 INQLDV-LOG-DEV-ID PIC 9(5).
05 INQLDV-RCODE PIC X(5).
01 OISCR-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 OISCR-FUNCTION PIC 9(4).
05 OISCR-RCODE PIC X(5).
01 OPNFRM-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 OPNFRM-RCODE PIC X(5).
01 OPNLDV-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 OPNLDV-LOG-DEV-ID PIC 9(5).
05 OPNLDV-RCODE PIC X(5).
01 OUTSCR-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 OUTSCR-RCODE PIC X(5).
01 PARFQN-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 PARFQN-PAR-NAME PIC X(120).
05 PARFQN-PAR-TYPE PIC X.
05 PARFQN-RCODE PIC X(5).
01 PDATA-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 PDATA-RCODE PIC X(5).
01 PUTATT-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 PUTATT-RCODE PIC X(5).
01 PUTBAK-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 PUTBAK-RCODE PIC X(5).
01 PUTCUR-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 PUTCUR-RCODE PIC X(5).
01 PUTLOC-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 PUTLOC-RCODE PIC X(5).
01 RMVPAG-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 RMVPAG-RCODE PIC X(5).
01 RPLFRM-FP-RECORD REDEFINES ADDELM-FP-RECORD.
05 RPLFRM-RCODE PIC X(5).
```

AI to FP message format, input parameters.

```
01 INPUT-RECORD.
05 ROUTINE-ID PIC 99.
01 ADDELM-FPAI-RECORD REDEFINES INPUT-RECORD.
05 ROUTINE-ID PIC 99.
05 ADDELM-ELEMENT-NAME PIC X(120).
```

```
01  ADDFRM-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  ADDFRM-WINDOW-NAME              PIC X(120).
    05  ADDFRM-FORM-NAME                PIC X(10).
01  CHGLDV-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  CHGLDV-LOG-DEV-ID              PIC 9(5).
01  CLSFRM-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  CLSFRM-FORM-NAME                PIC X(10).
01  CLSLDV-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  CLSLDV-LOG-DEV-ID              PIC 9(5).
01  GDATA-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  GDATA-INSTANCE-ID              PIC 9(4).
    05  GDATA-FIELD-NAME                PIC X(120).
01  GDATLN-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  GDATLN-FIELD-NAME                PIC X(120).
01  GETATT-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  GETATT-FIELD-NAME                PIC X(120).
    05  GETATT-DURATION                 PIC 9(4).
01  GETBAK-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  GETBAK-FIELD-NAME                PIC X(120).
    05  GETBAK-DURATION                 PIC 9(4).
01  GETCUR-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
01  GPAGE-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  GPAGE-WINDOW-NAME                PIC X(120).
    05  GPAGE-PAGE-NUMBER                PIC 9(4).
01  GWINDO-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  GWINDO-WINDOW-NAME                PIC X(120).
01  INQLDV-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
01  OISCR-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  OISCR-WINDOW-NAME                PIC X(120).
01  OPNFRM-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
    05  OPNFRM-FORM-NAME                PIC X(10).
01  OPNLDV-FPAI-RECORD REDEFINES INPUT-RECORD.
    05  ROUTINE-ID                      PIC 99.
```

```
01 OUTSCR-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 OUTSCR-WINDOW-NAME PIC X(120).
01 PARFQN-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 PARFQN-FIELD-NAME PIC X(120).
   05 PARFQN-FIELD-TYPE PIC X.
   05 PARFQN-LEVEL PIC 9(4).
01 PDATA-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 PDATA-FIELD-NAME PIC X(120).
   05 PDATA-BUFFER PIC X(4096).
01 PMSGLC-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 PMSGLC-MSG-CODE PIC X(5).
01 PMSGLS-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 PMSGLS-MSG-STRING PIC X(60).
01 PUTATT-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 PUTATT-FIELD-NAME PIC X(120).
   05 PUTATT-DURATION PIC 9(4).
   05 PUTATT-ATTRIBUTE PIC X(10).
01 PUTBAK-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 PUTBAK-FIELD-NAME PIC X(120).
   05 PUTBAK-DURATION PIC 9(4).
   05 PUTBAK-ATTRIBUTE PIC X(10).
01 PUTCUR-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 PUTCUR-FIELD-NAME PIC X(120).
01 PUTLOC-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 PUTLOC-FIELD-NAME PIC X(120).
   05 PUTLOC-ROW PIC 9(4).
   05 PUTLOC-COL PIC 9(4).
01 RMVPAG-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 RMVPAG-WINDOW-NAME PIC X(120).
   05 RMVPAG-PAGE-NUMBER PIC 9(4).
01 RPLFRM-FPAI-RECORD REDEFINES INPUT-RECORD.
   05 ROUTINE-ID PIC 99.
   05 RPLFRM-WINDOW-NAME PIC X(120).
   05 RPLFRM-PAGE-NUMBER PIC 9(4).
   05 RPLFRM-FORM-NAME PIC X(10).
```